

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

Claims 1-12 (Canceled).

13. (Original) A modular interconnection architecture for an expandable multiprocessor machine, based on a virtual bus hierarchy, comprising a given number of multiprocessor modules (QPi), each module including a plurality of processors and associated cache memories organized into nodes (Nj) and distributed on at least two interconnection levels: a first interconnection level (MI) corresponding to interconnection of the multiprocessor modules (QPi) within a node (Nj), and a second interconnection level (SI) corresponding to the interconnection of the nodes (Nj) with one another, the first interconnection level (MI) comprising connection agents (NCSi) connecting the multiprocessor modules (QPi) to one another and handling the transactions between the multiprocessor modules (QPi), the second interconnection level (SI) comprising external connection nodes (NCEj) connecting the nodes (Nj) to one another and handling the transactions between the nodes (Nj), the connection agents (NCSi) and the external connection nodes (NCEj) respectively having the same basic structure, the same external interface (XI), and adapted to implement the same coherency control protocol for the cache memories of the processors.

14. (Original) A modular interconnection architecture according to claim 13, characterized in that each external connection node (NCEj) comprises two identical connection agents (NCSi) connected head-to-tail, one of the two agents (NCSi) receiving and filtering transactions sent by the node (Nj) to which it is

5 connected, and the other agent (NCS"j) receiving and filtering the transactions sent by
6 the other nodes (Nj) to which it is connected.

1 15. (Original) A modular interconnection architecture according to claim
2 13, characterized in that each connection agent (NCSi) comprises an associative
3 memory (DDi) with a fixed size determined as a function of the number of processors
4 in the multiprocessor module (QPi) to which the connection agent (NCSi) is
5 connected, the state of the memories (DDi) being indicative of the presence of the last
6 modified data blocks in the cache memories of the multiprocessor module (QPi).

1 16. (Original) A modular interconnection architecture according to claim
2 14, characterized in that each connection agent (NCSi) comprises an associative
3 memory (DDi) with a fixed size determined as a function of the number of processors
4 in the multiprocessor module (QPi) to which the connection agent (NCSi) is
5 connected, the state of the memories (DDi) being indicative of the presence of the last
6 modified data blocks in the cache memories of the multiprocessor module (QPi).

1 17. (Original) A modular interconnection architecture according to claim
2 14, characterized in that the first and second head-to-tail connection agents (NCS'j
3 and NCS"j) only accept transactions for blocks modified in their respective
4 associative memories (DD'j and DD"j); modified data blocks in the first connection
5 agent (NCS'j) being exported to the requesting multiprocessor module or modules
6 and, conversely, modified data blocks in the second connection agent (NCS"j) being
7 imported from the module or modules holding the blocks.

1 18. (Original) A modular interconnection architecture according to claim
2 13, characterized in that the second interconnection level (SI) has a latency that is
3 double the latency of the first interconnection level (MI).

1 19. (Original) A modular interconnection architecture according to claim
2 14, characterized in that the second interconnection level (SI) has a latency that is
3 double the latency of the first interconnection level (MI).

1 20. (Original) A modular interconnection architecture according to claim
2 15, characterized in that the second interconnection level (SI) has a latency that is
3 double the latency of the first interconnection level (MI).

1 21. (Original) A modular interconnection architecture according to claim
2 16, characterized in that the second interconnection level (SI) has a latency that is
3 double the latency of the first interconnection level (MI).

1 22. (Original) A modular interconnection architecture according to claim
2 17, characterized in that the second interconnection level (SI) has a latency that is
3 double the latency of the first interconnection level (MI).

1 23. (Currently amended) A process for expanding the capacity of a
2 machine comprising a first given number of processors on a first level (MI) organized
3 into a first given number of multiprocessor modules (QPi) and capable of being
4 inserted into an interconnection architecture comprising a modular interconnection
5 architecture for an expandable multiprocessor machine, based on a virtual bus

6 hierarchy, comprising a given number of multiprocessor modules (QPi), each
 7 comprising a plurality of processors and associated cache memories organized into
 8 nodes (Nj) and distributed on at least two interconnection levels: a first
 9 interconnection level (MI) corresponding to interconnection of the multiprocessor
 10 modules (QPi) within a node (Nj), and a second interconnection level (SI)
 11 corresponding to the interconnection of the nodes (Nj) with one another, the first
 12 interconnection level (MI) comprising connection agents (NCSi) connecting the
 13 multiprocessor modules (QPi) to one another and handling the transactions between
 14 the multiprocessor modules (QPi), the second interconnection level (SI) comprising
 15 external connection nodes (NCEj) connecting the nodes (Nj) to one another and
 16 handling the transactions between the nodes (Nj), the connection agents (NCSi) and
 17 the external connection nodes (NCEj) respectively having the same basic structure,
 18 the same external interface (XI), and adapted to implement the same coherency
 19 control protocol for the cache memories of the processors, [[.]] characterized in that it
 20 consists of disconnecting one of the first-level multiprocessor modules (QPi) from its
 21 connection agent (NCSi) to free said connecting agent and of connecting, via said
 22 freed connection agent, a second given number of processors organized into a second
 23 given number of multiprocessor modules, also capable of being inserted into said
 24 interconnection architecture.

1 24. (Original) A process for expanding the capacity of a machine comprising
 2 a first given number of processors on a first level (MI) organized into a first given
 3 number of multiprocessor modules (QPi) and capable of being inserted into an
 4 interconnection architecture comprising a modular interconnection architecture for an
 5 expandable multiprocessor machine, based on a virtual bus hierarchy, comprising a

6 given number of multiprocessor modules (Q_{Pi}), each comprising a plurality of
 7 processors and associated cache memories organized into nodes (N_j) and distributed
 8 on at least two interconnection levels: a first interconnection level (MI) corresponding
 9 to interconnection of the multiprocessor modules (Q_{Pi}) within a node (N_j), and a
 10 second interconnection level (SI) corresponding to the interconnection of the nodes
 11 (N_j) with one another, the first interconnection level (MI) comprising connection
 12 agents (NCS_i) connecting the multiprocessor modules (Q_{Pi}) to one another and
 13 handling the transactions between the multiprocessor modules (Q_{Pi}), the second
 14 interconnection level (SI) comprising external connection nodes (NCE_j) connecting
 15 the nodes (N_j) to one another and handling the transactions between the nodes (N_j),
 16 the connection agents (NCS_i) and the external connection nodes (NCE_j) respectively
 17 having the same basic structure, the same external interface (XI), and adapted to
 18 implement the same coherency control protocol for the cache memories of the
 19 processors, each external connection node (NCE_j) comprises two identical connection
 20 agents (NCS_i) connected head-to-tail, one of the two agents (NCS'_j) receiving and
 21 filtering transactions sent by the node (N_j) to which it is connected, and the other
 22 (NCS''_j) receiving and filtering the transactions sent by the other nodes (N_j) to which
 23 it is connected, characterized in that it consists of disconnecting one of the first-level
 24 multiprocessor modules (Q_{Pi}) from its connection agent (NCS_i) to free said
 25 connecting agent and of connecting, via said freed connection agent, a second given
 26 number of processors organized into a second given number of multiprocessor
 27 modules, also capable of being inserted into said interconnection architecture.

1 25. (Original) A process for expanding the capacity of a machine
 2 comprising a first given number of processors on a first level (MI) organized into a

3 first given number of multiprocessor modules (Q_{Pi}) and capable of being inserted into
4 an interconnection architecture comprising a modular interconnection architecture for
5 an expandable multiprocessor machine, based on a virtual bus hierarchy, comprising a
6 given number of multiprocessor modules (Q_{Pi}), each comprising a plurality of
7 processors and associated cache memories organized into nodes (N_j) and distributed
8 on at least two interconnection levels: a first interconnection level (MI) corresponding
9 to interconnection of the multiprocessor modules (Q_{Pi}) within a node (N_j), and a
10 second interconnection level (SI) corresponding to the interconnection of the nodes
11 (N_j) with one another, the first interconnection level (MI) comprising connection
12 agents (NCS_i) connecting the multiprocessor modules (Q_{Pi}) to one another and
13 handling the transactions between the multiprocessor modules (Q_{Pi}), the second
14 interconnection level (SI) comprising external connection nodes (NCE_j) connecting
15 the nodes (N_j) to one another and handling the transactions between the nodes (N_j),
16 the connection agents (NCS_i) and the external connection nodes (NCE_j) respectively
17 having the same basic structure, the same external interface (XI), and adapted to
18 implement the same coherency control protocol for the cache memories of the
19 processors, characterized in that it consists of disconnecting one of the first-level
20 multiprocessor modules (Q_{Pi}) from its connection agent (NCS_i) to free said
21 connecting agent and of connecting, via said freed connection agent, a second given
22 number of processors organized into a second given number of multiprocessor
23 modules, also capable of being inserted into said interconnection architecture, each
24 connection agent (NCS_i) comprises an associative memory (DD_i) with a fixed size
25 determined as a function of the number of processors in the multiprocessor module
26 (Q_{Pi}) to which the connection agent (NCS_i) is connected, the state of the memories

27 (DDi) being indicative of the presence of the last modified data blocks in the cache
 28 memories of the multiprocessor module (QPi).

1 26. (Original) A process according to claim 24, characterized in the
 2 connection agents (NCS'j) and (NCS''j) respectively comprise an associative memory
 3 (DD'j) and (DD''j) with a fixed size determined as a function of the number of
 4 processors in the multiprocessor module (HBj) to which the connection agents
 5 (NCS'j) and (NCS''j) are connected, the state of the associated memories (DD'j) and
 6 (DD''j) being indicative of the presence of the last modified data blocks executed or
 7 conversely imported.

1 27. (Currently amended) A process for expanding the capacity of a machine
 2 according to claim 24 wherein the first and second head-to-tail connection agents
 3 (NCS'j and NCS''j) only accept transactions for blocks modified in their respective
 4 associative memories (DD'j and DD''j); modified data blocks in the first connection
 5 agent (NCS'j) being exported to the requesting multiprocessor module or modules
 6 and, conversely, modified data blocks in the second connection agent (NCS''j) being
 7 imported from the module or modules holding the blocks, ~~characterized in that it~~
 8 ~~consists of disconnecting one of the first level multiprocessor modules (QPi) from its~~
 9 ~~connection agent (NCSi) to free said connecting agent and of connecting, via this~~
 10 ~~freed connection agent, a second given number of processors organized into a second~~
 11 ~~given number of multiprocessor modules, also capable of being inserted into said~~
 12 ~~interconnection architecture comprising a modular interconnection architecture.~~

1 28. (Currently amended) A process for expanding the capacity of a
 2 machine according to claim 24 wherein the second interconnection level (SI) has a
 3 latency that is double the latency of the first interconnection level (MI), ~~characterized~~
 4 ~~in that it consists of disconnecting one of the first level multiprocessor modules (QPi)~~
 5 ~~from its connection agent (NCSi) to free said connecting agent and of connecting, via~~
 6 ~~this freed connection agent, a second given number of processors organized into a~~
 7 ~~second given number of multiprocessor modules, also capable of being inserted into~~
 8 ~~said interconnection architecture.~~

1 29. (Original) A process according to claim 23, characterized in that, the
 2 second given number of processors being organized into a second given number of
 3 multiprocessor modules on a second level (SI), it consists of connecting it to the
 4 connection agent (NCSi) of the first given number of processors on the first level (MI)
 5 through one of the connection agents (NCS"j) on the second level.

1 30. (Original) A process according to claim 29, characterized in that, the
 2 second given number of processors also being on the first level (MI), and further
 3 comprising connecting the respective connection agents (NCS') to the first and second
 4 given numbers of processors, the second level (SI) being reduced to a single link.

1 31. (Currently amended) An expandable ~~multi-node~~multinode
 2 multiprocessor machine, comprising an interconnection architecture including a given
 3 number of multiprocessor modules including a modular interconnection architecture
 4 for an expandable multiprocessor machine, based on a virtual bus hierarchy,
 5 comprising a given number of multiprocessor modules (QPi) , each module including

6 a plurality of processors and associated cache memories organized into nodes (N_j) and
 7 distributed on at least two interconnection levels: a first interconnection level (MI)
 8 corresponding to interconnection of the multiprocessor modules (Q_{Pi}) within a node
 9 (N_j), and a second interconnection level (SI) corresponding to the interconnection of
 10 the nodes (N_j) with one another, the first interconnection level (MI) comprising
 11 connection agents (NCS_i) connecting the multiprocessor modules (Q_{Pi}) to one
 12 another and handling the transactions between the multiprocessor modules (Q_{Pi}), the
 13 second interconnection level (SI) comprising external connection nodes (NCE_j)
 14 connecting the nodes (N_j) to one another and handling the transactions between the
 15 nodes (N_j), the connection agents (NCS_i) and the external connection nodes (NCE_j)
 16 respectively having the same basic structure, the same external interface (XI), and
 17 adapted to implement the same coherency control protocol for the cache memories of
 18 the processors.

1 32. (Original) An expandable multinode multiprocessor machine as set
 2 forth in claim 31, characterized in that each external connection node (NCE_j)
 3 comprises two identical connection agents (NCS_i) connected head-to-tail, one of the
 4 two agents (NCS'_j) receiving and filtering transactions sent by the node (N_j) to which
 5 it is connected, and the other agent (NCS''_j) receiving and filtering the transactions
 6 sent by the other nodes (N_j) to which it is connected.

1 33. (Original) An expandable multinode multiprocessor machine as set
 2 forth in claim 31, characterized in that each connection agent (NCS_i) comprises an
 3 associative memory (DD_i) with a fixed size determined as a function of the number of
 4 processors in the multiprocessor module (Q_{Pi}) to which the connection agent (NCS_i)

5 is connected, the state of the memories (DDi) being indicative of the presence of the
 6 last modified data blocks in the cache memories of the multiprocessor module (QPi).

1 34. (Original) An expandable multinode multiprocessor machine as set
 2 forth in claim 31, characterized in that each connection agent (NCSi) comprises an
 3 associative memory (DDi) with a fixed size determined as a function of the number of
 4 processors in the multiprocessor module (QPi) to which the connection agent (NCSi)
 5 is connected, the state of the memories (DDi) being indicative of the presence of the
 6 last modified data blocks in the cache memories of the multiprocessor module (QPi).

1 35. (Original) An expandable multinode multiprocessor machine as set
 2 forth in claim 31, characterized in that the first and second head-to-tail connection
 3 agents (NCS'j and NCS''j) only accept transactions for blocks modified in their
 4 respective associative memories (DD'j and DD''j); modified data blocks in the first
 5 connection agent (NCS'j) being exported to the requesting multiprocessor module or
 6 modules and, conversely, modified data blocks in the second connection agent
 7 (NCS''j) being imported from the module or modules holding the blocks.

1 36. (Original) An expandable multinode multiprocessor machine as set
 2 forth in claim 31, characterized in that the second interconnection level (SI) has a
 3 latency that is double the latency of the first interconnection level (MI).

1 37 (Currently amended) A process for tracing data blocks in an
 2 interconnection architecture for an expandable microprocessor machine, based on a
 3 virtual bus hierarchy, comprising a given number of multiprocessor modules (QPi)

4 ,each module including a plurality of processors and associated cache memories
 5 organized into nodes (Nj) and distributed on at least two interconnection levels: a first
 6 interconnection level (MI) corresponding to interconnection of the multiprocessor
 7 modules (QPi) within a node (Nj), and a second interconnection level (SI)
 8 corresponding to the interconnection of the nodes (Nj) with one another, the first
 9 interconnection level (MI) comprising connection agents (NCSi) connecting the
 10 multiprocessor modules (QPi) to one another and handling the transactions between
 11 the multiprocessor modules (QPi), the second interconnection level (SI) comprising
 12 external connection nodes (NCEj) connecting the nodes (Nj) to one another and
 13 handling the transactions between the nodes (Nj), the connection agents (NCSi) and
 14 the external connection nodes (NCEj) respectively having the same basic structure,
 15 the same external interface (XI), and adapted to ~~implement~~implement the same
 16 coherency control protocol for the cache memories of the processors, comprising
 17 duplicating on the first level in the associative memories (DDi) only modified data
 18 blocks in the cache memories of the multiprocessor modules (QPi) and tracing only
 19 the modified blocks inside the node (Nj).

1 38. (Original) A process for tracing data blocks as set forth in claim 37
 2 wherein each external connection node (NCEj) comprises two identical connection
 3 agents (NCSi) connected head-to-tail, one of the two agents (NCS'j) receiving and
 4 filtering transactions sent by the node (Nj) to which it is connected, and the other
 5 agent (NCS''j) receiving and filtering the transactions sent by the other nodes (Nj) to
 6 which it is connected.

1 39. (Original) A process for tracing data blocks as set forth in claim 37
2 wherein each connection agent (NCS_i) comprises an associative memory (DD_i) with a
3 fixed size determined as a function of the number of processors in the multiprocessor
4 module (QP_i) to which the connection agent (NCS_i) is connected, the state of the
5 memories (DD_i) being indicative of the presence of the last modified data blocks in
6 the cache memories of the multiprocessor module (QP_i).

1 40. (Original) A process for tracing data blocks as set forth in claim 38,
2 characterized in that the first and second head-to-tail connection agents (NCS'_j and
3 NCS''_j) only accept transactions for blocks modified in their respective associative
4 memories (DD'_j and DD''_j); modified data blocks in the first connection agent (NCS'_j)
5 being exported to the requesting multiprocessor module or modules and, conversely,
6 modified data blocks in the second connection agent (NCS''_j) being imported from the
7 module or modules holding the blocks

1 41. (Original) A process for tracing data blocks as set forth in claim 38,
2 characterized in that the first and second head-to-tail connection agents (NCS'_j and
3 NCS''_j) only accept transactions for blocks modified in their respective associative
4 memories (DD'_j and DD''_j); modified data blocks in the first connection agent (NCS'_j)
5 being exported to the requesting multiprocessor module or modules and, conversely,
6 modified data blocks in the second connection agent (NCS''_j) being imported from the
7 module or modules holding the blocks.

1 42. (Currently amended) A process for tracing data blocks as set forth in
 2 claim 1337, characterized in that the second interconnection level (SI) has a latency
 3 that is double the latency of the first interconnection level (MI).

1 43. (Original) A process for tracing data blocks in a modular
 2 interconnection architecture for an expandable multiprocessor machine, based on a
 3 virtual bus hierarchy, comprising a given number of multiprocessor modules (QPi) ,
 4 each module including a plurality of processors and associated cache memories
 5 organized into nodes (Nj) and distributed on at least two interconnection levels: a first
 6 interconnection level (MI) corresponding to interconnection of the multiprocessor
 7 modules (QPi) within a node (Nj), and a second interconnection level (SI)
 8 corresponding to the interconnection of the nodes (Nj) with one another, the first
 9 interconnection level (MI) comprising connection agents (NCSi) connecting the
 10 multiprocessor modules (QPi) to one another and handling the transactions between
 11 the multiprocessor modules (QPi), the second interconnection level (SI) comprising
 12 external connection nodes (NCEj) connecting the nodes (Nj) to one another and
 13 handling the transactions between the nodes (Nj), the connection agents (NCSi) and
 14 the external connection nodes (NCEj) respectively having the same basic structure,
 15 the same external interface (XI), and adapted to implement the same coherency
 16 control protocol for the cache memories of the processors, characterized in that it
 17 consists, on the second level (SI), of duplicating in the associative memories (DD'j
 18 and DD"j) of the connection agents (NCS' and NCS"j) of each external connection
 19 node (NCEj) only the modified blocks exported, or conversely imported, and of
 20 tracing only the modified blocks exported, or conversely imported, between each
 21 node (Nj) of the machine.

1 44. (Original) A process for tracing data blocks as set forth in claim 43,
2 wherein each external connection node (NCE_j) comprises two identical connection
3 agents (NCS_i) connected head-to-tail, one of the two agents (NCS'_j) receiving and
4 filtering transactions sent by the node (N_j) to which it is connected, and the other
5 agent (NCS''_j) receiving and filtering the transactions sent by the other nodes (N_j) to
6 which it is connected.

1 45. (Original) A process for tracing data blocks as set forth in claim 43,
2 wherein each connection agent (NCS_i) comprises an associative memory (DD_i) with a
3 fixed size determined as a function of the number of processors in the multiprocessor
4 module (QPi) to which the connection agent (NCS_i) is connected, the state of the
5 memories (DD_i) being indicative of the presence of the last modified data blocks in
6 the cache memories of the multiprocessor module (QPi).

1 46. (Original) A process for tracing data blocks as set forth in claim 44,
2 wherein each connection agent (NCS_i) comprises an associative memory (DD_i) with a
3 fixed size determined as a function of the number of processors in the multiprocessor
4 module (QPi) to which the connection agent (NCS_i) is connected, the state of the
5 memories (DD_i) being indicative of the presence of the last modified data blocks in
6 the cache memories of the multiprocessor module (QPi).

1 47. (Original) A process for tracing data blocks as set forth in claim 44,
2 wherein the first and second head-to-tail connection agents (NCS'_j and NCS''_j) only
3 accept transactions for blocks modified in their respective associative memories (DD'_j
4 and DD''_j); modified data blocks in the first connection agent (NCS'_j) being exported

5 to the requesting multiprocessor module or modules and, conversely, modified data
6 blocks in the second connection agent (NCS"j) being imported from the module or
7 modules holding the blocks.

1 48. (Original) A process for tracing data blocks as set forth in claim 43,
2 wherein the second interconnection level (SI) has a latency that is double the latency
3 of the first interconnection level (MI).